

(12) DEMANDE INTERNATIONALE PUBLIÉE EN VERTU DU TRAITÉ DE COOPÉRATION
EN MATIÈRE DE BREVETS (PCT)

(19) Organisation Mondiale de la Propriété
Intellectuelle
Bureau international



(43) Date de la publication internationale
31 octobre 2002 (31.10.2002)

PCT

(10) Numéro de publication internationale
WO 02/087191 A2

(51) Classification internationale des brevets⁷ : H04L 29/06

(21) Numéro de la demande internationale :

PCT/FR02/01365

(22) Date de dépôt international : 22 avril 2002 (22.04.2002)

(25) Langue de dépôt :

français

(26) Langue de publication :

français

(30) Données relatives à la priorité :

01/05464 23 avril 2001 (23.04.2001) FR

(71) Déposant (pour tous les États désignés sauf US) : IN-
TOAN TECHNOLOGY [FR/FR]; 2, rue du Chemin Vert,
F-92110 Clichy (FR).

(72) Inventeurs; et

(75) Inventeurs/Déposants (pour US seulement) :
DUQUEENNE, Jonathan [FR/FR]; 2, rue du Chemin
Vert, F-92110 Clichy (FR). BONAN, Fabrice [FR/FR]; 2,
rue du Chemin Vert, F-92110 Clichy (FR).

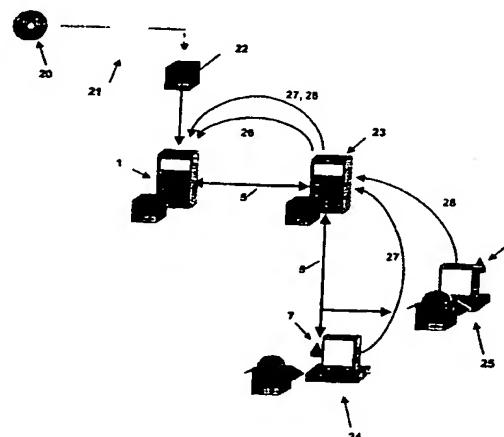
(74) Mandataire : LEBOYER, Jean-Jacques; Cabinet
Leboyer, 12, rue du Helder, F-75009 Paris (FR).

(81) États désignés (national) : AE, AG, AL, AM, AT, AU, AZ,
BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ,
DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM,
HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK,
LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX,
MZ, NO, NZ, OM, PH, PL, PT, RO, RU, SD, SE, SG, SI,

[Suite sur la page suivante]

(54) Title: SYSTEM AND METHOD FOR DYNAMIC DISTRIBUTION OF DATA AND/OR SERVICES

(54) Titre : SYSTÈME ET PROCÉDÉ POUR LA DISTRIBUTION DYNAMIQUE DE DONNÉES ET/OU DE SERVICES



WO 02/087191 A2

(57) Abstract: The invention concerns the dynamic distribution of data and/or services. The system used comprises essentially: an IRC server (1), and optionally secondary servers (6); at least an agent (7), resident or capable of becoming resident on a user terminal, and adapted to be linked in communication with the server (1), and/or at least a secondary server software (6); and at least a communication layer (18) designed to transport remote method calls on objects located on a server (1) and/or secondary servers (6), and in implementing at least a communication protocol (5) ensuring an interactive link between a server (1, 6) and agents (7), while said protocol comprises: means enabling it to produce and manage communications between at least an agent and IRC, in particular through XML messages; means for calling a remote method by reading/writing of objects respectively of the types *ObjectRequest* or and or *ObjectResponse*. The invention is applicable on all types of distributed networks and architectures, with traceability.

(57) Abrégé : L'invention concerne la distribution dynamique de données et/ou de services. Le système mis en oeuvre comporte essentiellement: un serveur IRC (1), et éventuellement des serveurs fils (6); au moins un agent (7), résident ou apte à devenir résident sur un terminal utilisateur,

[Suite sur la page suivante]



SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VN,
YU, ZA, ZM, ZW.

(84) **États désignés (régional) :** brevet ARIPO (GH, GM, KE,
LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), brevet
eurasien (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), brevet
européen (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR,
IE, IT, LU, MC, NL, PT, SE, TR), brevet OAPI (BF, BJ,
CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN,
TD, TG).

Publiée :

— *sans rapport de recherche internationale, sera publiée
dès réception de ce rapport*

*En ce qui concerne les codes à deux lettres et autres abrévia-
tions, se référer aux "Notes explicatives relatives aux codes et
abréviations" figurant au début de chaque numéro ordinaire de
la Gazette du PCT.*

et apte à être mis en communication avec le serveur (1), et/ou au moins un logiciel serveur fils (6); et au moins une couche de communication (18) apte à véhiculer des appels de méthodes distantes sur des objets situés sur un serveur (1) et/ou un serveur fils (6), et à mettre en oeuvre au moins un protocole de communication (5) apte à assurer un lien interactif entre un serveur (1,6) et des agents (7), tandis que ledit protocole comporte: des moyens le rendant apte à produire et gérer des communications entre au moins un agent et l'IRC, notamment par le biais de messages XML; des moyens pour appeler une méthode distante par lecture/écriture d'objets de types respectivement *Object/Request* ou *ObjetRequête* et *ObjectResponse* ou *ObjetRéponse*. Application sur tous types de réseaux et d'architectures distribuées, avec traçabilité.

SYSTÈME ET PROCÉDÉ POUR LA DISTRIBUTION DYNAMIQUE DE DONNÉES ET/OU DE SERVICES

5 DOMAINE DE L'INVENTION

La présente invention concerne le domaine de la distribution d'applications et de services à des utilisateurs distants. Elle concerne plus particulièrement la distribution d'applications et/ou de services, notamment la distribution dynamique de données, à des utilisateurs distants connectés, 10 sur tous types de réseaux et d'architectures distribuées.

Les données sont des biens numériques et seront qualifiées dans la suite aussi bien par l'une que par l'autre de ces dénominations.

ARRIÈRE-PLAN TECHNOLOGIQUE DE L'INVENTION

15 On connaît des systèmes de distribution et/ou de vérification de données en réseau, dont certains comportent une distribution sécurisée, visant à procurer des solutions ponctuelles, spécifiques pour chaque cas.

Ainsi, la demande de brevet WO 99/50995 décrit un intégrateur de services Internet composé d'une plate-forme qui choisit, intègre et configure 20 automatiquement, selon les demandes du client, les fonctions nécessaires à la mise en place d'un service Internet à valeur ajoutée.

Le document US 6075943 décrit un système pour le transfert et l'installation à distance de programmes d'application client-serveur à partir d'un ordinateur source, dans un système de traitement de données distant. 25 Un cycle de distribution peut intégrer plusieurs programmes pour des destinataires différents ou pour un même destinataire. Ce système utilise un programme de décompression qui doit être transféré avant les programmes d'application comprimés à transférer. De plus, les erreurs sont détectées, mais non traitées automatiquement.

30 Le document US 6141759 décrit un système de gestion des requêtes d'information sur un réseau d'ordinateurs. Il met en oeuvre un ou plusieurs systèmes d'ordinateurs serveurs secondaires, dont la fonction est de se suppléer en cas de besoin à un premier ordinateur serveur. Les erreurs éventuellement détectées dans le traitement de la requête d'information 35 donnent lieu à une réinitialisation du processus de traitement.

Le document US 5999741 décrit une méthode d'installation à distance d'une mise à jour de logiciel, dans laquelle le système mis en oeuvre sélectionne une voie de communication pour requérir des informations sur la version existante et télécharger la mise à jour du logiciel.

5 La présente invention vise à procurer à des utilisateurs distants, ainsi qu'aux fournisseurs avec lesquels ils sont connectés, des moyens de distribution dynamique d'applications et de services, sous forme de flux d'information numérisée, de préférence sécurisée, et avantageusement avec une traçabilité permettant, par exemple de connaître l'utilisation d'un objet 10 et/ou de localiser un objet, et au besoin d'obtenir des profils utilisateurs.

RÉSUMÉ DE L'INVENTION

Le système, le procédé et les moyens selon l'invention reposent sur une architecture client/serveur. L'invention met en oeuvre une plate-forme 15 de communication centrale ou serveur (encore dénommé Resource Center, ou IRC), qui dirige des agents aptes à diffuser les données vers les postes finals (ou postes finaux) ou vers des Resource Centers (logiciels serveurs) secondaires (encore dénommés Resource Centers fils), servant de relais vers de tels agents, tandis que les communications entre la plate-forme de 20 communication centrale et/ou les centres de ressource secondaires et les agents utilisent avantageusement un protocole sécurisé.

Le système selon l'invention comporte une modularité et des moyens de traçabilité qui en font un outil particulièrement adapté pour des transmissions dynamiques de données et de services. En particulier pour les 25 fournisseurs de services professionnels en ligne (NSP, ASP, MSP), il permet un positionnement sur de nouveaux marchés et le développement d'offres, originales, avec une qualité de service exceptionnelle, non accessible aux systèmes préexistants. Il améliore en effet la performance des offres de services en ligne en assurant :

30

- la mise à disposition et la fiabilité des services proposés,
- la sécurité et la confidentialité des données échangées, et
- la simplicité d'utilisation et l'accessibilité des services.

Le système selon l'invention pour la distribution d'applications et de services à des utilisateurs distants comporte principalement au moins une 35 plate-forme de communication centrale (IRC), à laquelle sont reliés

fonctionnellement des agents permettant la diffusion de données vers les postes utilisateurs finals et/ou, en option, des centres de ressource fils agencés pour servir eux-mêmes de relais vers lesdits agents.

La communication entre la plate-forme de communication (dénommée ici Intoan Resource Center, soit en abrégé IRC) et ses agents met avantageusement en oeuvre un protocole dénommé en abrégé IVAP (Intoan Versatile Access Protocol), qui est un protocole sécurisé élaboré par la demanderesse.

10 L'invention procure ainsi un système permettant de produire et de gérer des flux de biens numériques, ainsi que de l'information associée à ceux-ci, avantageusement au travers de l'ensemble des acteurs, depuis la création du bien jusqu'à sa consommation, avec notamment pour objectif de maximiser la valeur de ce bien telle qu'elle est perçue par un client final.

15 Le bien créé peut être d'un type préexistant chez le fournisseur ou être créé spécialement à la demande.

L'invention est décrite ci-après plus en détail, en référence aux dessins annexés, qui ne la limitent aucunement et sont uniquement illustratifs de certains modes de réalisation de l'invention.

20 BRÈVE DESCRIPTION DES DESSINS

L'invention est illustrée ci-après plus en détail, en référence aux dessins annexés, dans lesquels:

- FIG. 1 représente schématiquement les éléments constitutifs du système selon l'invention et de son environnement;
- 25 - FIG. 2 est une représentation schématique d'une forme de réalisation d'une architecture modulaire commune au serveur et à un agent;
- FIG. 3 représente une forme de réalisation du protocole IVAP;
- FIG. 4 illustre la traçabilité des biens numériques et des procédures dans un exemple de distribution dynamique selon l'invention; et
- 30 - FIG. 5 montre un exemple de diagramme de séquence représentant le fonctionnement du protocole de communication avantageusement mis en oeuvre selon l'invention dans une application client/serveur.

Dans les figures annexées, les mêmes numéros de référence sont, sauf exceptions, utilisés pour les mêmes éléments apparaissant sur plusieurs figures.

DESCRIPTION DÉTAILLÉE DE L'INVENTION

Bien que la terminologie utilisée dans la présente demande soit classique et/ou aisément compréhensible par l'homme du métier des différents domaines conceptuels, techniques et opérationnels concernés, les définitions de certains termes seront fournies plus loin pour en garantir la clarté. Certains noms utilisés dans la suite sont des dénominations commerciales appartenant à leurs titulaires respectifs.

La présente invention a pour premier objet un système pour la distribution dynamique de données et/ou de services, comportant essentiellement:

- un serveur (IRC), et éventuellement des serveurs fils (ou Resource Centers fils);
- au moins un agent, c'est-à-dire un support de programme, celui-ci pouvant être mis en communication avec le serveur et/ou le(s) Resource Center(s) fils, et étant apte à devenir résident sur la machine de l'utilisateur;
- au moins une couche de communication apte à véhiculer des appels de méthodes distantes sur des objets situés sur un serveur et/ou un serveur fils, et à mettre en oeuvre au moins un protocole de communication apte à assurer un lien interactif entre un serveur et des agents, tandis que ledit protocole comporte:
 - des moyens le rendant apte à produire et gérer des communications entre au moins un agent et l'IRC, notamment par le biais de messages XML;
 - des moyens pour appeler une méthode distante par lecture/écriture d'objets de types respectivement *ObjectRequest* (ObjetRequête) et *ObjectResponse* (ObjetRéponse), avantageusement au moyen d'un *stub* et d'un *skeleton*.

Ainsi, si l'on se réfère à la FIG. 1, une forme de réalisation du système selon l'invention comporte un serveur IRC père 1, intégrant des services 2, mis en communication interactive par une liaison filaire 3 et/ou une liaison sans fil 4, au moyen d'un protocole 5 (ici protocole IVAP) avec, par exemple un serveur fils 6 et/ou au moins un agent 7, résident ou devenu résident sur un terminal utilisateur tel qu'un téléphone 8, un PDA 9, une boîte dite Set

Top Box 10 et/ou un terminal informatique 11 PC, Mac[®] ou serveur Sun[®], ou autre. En option, un ou plusieurs serveurs IRC fils 6 peuvent également être présents.

5 Le système selon l'invention peut fonctionner sur tous types de réseaux. Il permet avantageusement le transport sécurisé de tous types de contenus (par exemple textes, images, logiciels, vidéo, etc.) à destination de tous types de postes terminaux, tels que par exemple des ordinateurs, des téléphones, des assistants numériques, entre autres.

10 Dans ce système, le ou les serveurs et les agents sont construits autour de la même architecture. Dans cette architecture, les couches de base intégrées tant dans le serveur que dans les agents et/ou dans les Resource Centers fils comportent des moyens et/ou des fonctionnalités appropriées pour assurer:

15

- la communication avec les agents (couche IVAP): échanges de messages et échanges binaires, et en option,
- le cryptage (par l'intermédiaire d'une couche SSL par exemple), et/ou
- la compression (par l'intermédiaire d'une couche Zlib par exemple), et/ou
- 20 - l'ordonnancement des tâches, et/ou
- l'accès à des bases de données externes, et/ou
- des ajouts/suppressions de modules, notamment.

Les couches de base sont avantageusement intégrées dans un "framework", qui permet l'ajout dynamique de modules.

25 La FIG. 2 représente ainsi schématiquement une architecture modulaire, qui est commune au serveur et à l'agent, comportant une couche de cryptage 12, une couche de communication apte à mettre en oeuvre un protocole IVAP 5, des core modules (ou modules centraux) 13,14, des modules additionnels 15,16, les uns et les autres en nombre indéfini, ainsi 30 qu'un système de gestion de base de données 17, ce dernier étant optionnel pour l'agent 7 (voir FIG. 1).

35 Lesdits modules sont par exemple, isolément ou cumulativement des modules du type visant à permettre le déploiement d'applications, la surveillance de machines, la maintenance à distance, la téléconfiguration d'un poste de travail informatisé, la télémaintenance de progiciels, la

sauvegarde des données utilisateur, entre autres fonctionnalités et/ou applications.

Le logiciel serveur, dénommé en abrégé IRC dans sa forme de réalisation préférée, est de préférence développé en langage Java.

5 L'implémentation de l'agent pour le système d'exploitation d'un terminal connecté (par exemple un agent pour Windows®) est de préférence effectuée en C++. L'agent peut cependant être développé en n'importe quel langage, et ce en fonction des architectures. L'homme de l'art est apte à inventorier ces situations, à les analyser et à sélectionner un ou plusieurs 10 langages appropriés pour une architecture définie.

A titre d'exemple non limitatif, la FIG. 3 illustre une architecture d'éléments de protocole d'un système selon l'invention. Dans cette figure sont schématisés une couche de cryptage 12 (couche SSL par exemple), le protocole de communication 5 IVAP, un canal XML 18 pour l'échange des 15 messages, et un canal binaire 19 pour l'échange de données.

Un agent, au sens où on l'entend selon la présente invention, est un support de programme, qui peut être acquis en tant que tel sous forme de 20 cédérom, DVD ou autre, ou formaté et chargé à partir d'un site Internet, entre autres, l'acquéreur choisissant l'agent correspondant à l'architecture de sa machine, qu'il s'agisse d'un PC, d'un Mac®, d'un Palm®, ou d'autres appareils destinés à héberger ledit agent, et qui comporte avantageusement au moins un programme souhaité ou sur lequel au moins un programme peut être enregistré, avantageusement par téléchargement ou télétransfert, par exemple à partir d'un site Internet fournisseur.

25 La machine intégrant ou recevant l'agent support de programme peut être indifféremment un ordinateur PC ou Mac, un assistant personnel, un téléphone mobile, ou tout appareil analogue.

A la fin de son installation, l'agent devient résident sur la machine.

Dans une forme de réalisation préférée, le système peut être 30 totalement ou partiellement autonome, dynamique et évolutif. Il peut conférer à l'utilisateur final une réelle liberté, dans la mesure où cet utilisateur final n'a pas nécessairement à intervenir dans le procédé de gestion ou de maintenance, et peut travailler indépendamment du système, qui fonctionne avec transparence.

Un tel système conforme à l'invention peut être utilisé tant au sein de "grands comptes" qu'au sein de TPE/PME (c'est-à-dire des très petites entreprises ou des petites et moyennes entreprises).

5 Le système selon l'invention est avantageusement mis en oeuvre selon un procédé qui constitue un autre objet de la présente invention.

L'invention a ainsi également pour objet un procédé pour la distribution dynamique de données et/ou de services, comportant essentiellement les étapes consistant à:

- 10 - procurer au moins un agent, c'est-à-dire un support de programme, qui peut être acquis en tant que tel sous forme de cédérom, DVD ou autre, ou formaté et chargé à partir d'un site Internet, entre autres. L'acquéreur choisit l'agent correspondant à l'architecture de sa machine, qu'il s'agisse d'un PC, d'un Mac®, d'un Palm®, ou d'autres appareils destinés à héberger ledit agent;
- 15 - installer un tel agent, pour qu'il devienne résident sur la machine de l'utilisateur;
- effectuer la souscription de l'agent par l'utilisateur aux services proposés, directement depuis l'interface de l'agent par l'intermédiaire d'un annuaire, et ce aussi bien pour des services locaux (c'est-à-dire définis sur le 20 Resource Center local) que pour des services publics, et aussi bien pour des services payants que pour des services gratuits;
- en option, effectuer l'abonnement de l'agent aux services par l'administrateur du système depuis l'interface de l'IRC (c'est à dire de façon centralisée);
- 25 - effectuer l'inscription à ces services, avec corrélativement chargement ou téléchargement automatique des composants nécessaires à leur activation.

30 L'édit protocole est destiné à optimiser les transferts (compression, utilisation de la bande passante, et autres moyens) entre les IRC et les agents. Il peut être activé sans que l'on ait à accroître la vitesse des échanges, qui demeure fonction de la qualité du réseau sélectionné.

35 Dans une forme de réalisation avantageuse, le procédé selon l'invention met en oeuvre un protocole, avantageusement le protocole IVAP susdit (voir FIG. 3), dont les étapes essentielles consistent à:

- produire et gérer des communications entre au moins un agent et l'IRC, notamment par le biais de messages XML;
- appeler au moins une méthode distante par lecture/écriture d'objets de types respectivement *ObjectRequest* (ObjetRequête) et 5 *ObjectResponse* (ObjetRéponse), avantageusement au moyen d'un *stub* et d'un *skeleton*.

Plus précisément, l'agent 7 communique avec l'IRC 1 par le biais de messages XML. La couche de communication XML 18 sert, la plupart du temps, à véhiculer des appels de méthodes distantes sur des objets situés sur 10 le serveur Resource Center.

15 Selon une forme de fonctionnement préférée, l'appel d'une méthode distante s'effectue par lecture/écriture d'objets respectivement *ObjectRequest* et *ObjectResponse*.

Un *ObjectRequest* encapsule un appel de méthode, c'est-à-dire qu'il 15 contient:

le nom de la méthode distante (une simple chaîne de caractères) et
une liste de paramètres (c'est-à-dire d'autres objets).

Un *ObjectResponse* encapsule l'état de retour de la méthode distante, c'est-à-dire, par exemple:

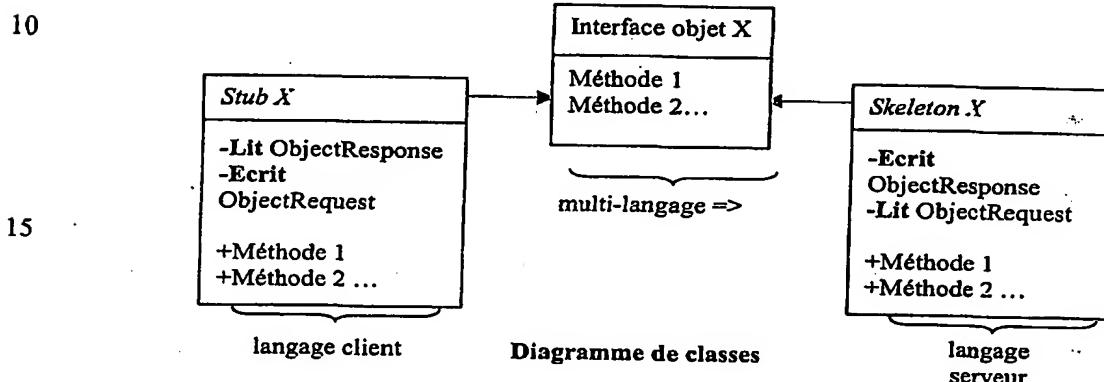
20

- un objet valeur de retour,
- un objet exception (si une exception, levée à l'exécution côté serveur, doit être interceptée par le client),
- le (ou les) objet(s) passé(s) par référence à la méthode dont l'état doit changer.

25 Le processus de lecture/écriture d'un objet dans un flux de données (réseau, fichier, etc.) correspond au processus de sérialisation (écriture) et de désérialisation (lecture), et permet de rendre les objets persistants, c'est-à-dire matérialisables sur une plate-forme distante. La communication par flux d'objets préconisée selon l'invention apporte le niveau d'abstraction nécessaire au traitement aisés des échanges entre client et serveur. La 30 formalisation des processus de sérialisation/désérialisation permet d'en automatiser la mise en oeuvre, c'est-à-dire la programmation.

35 L'appel de méthode distante utilise ce que l'on nomme classiquement un *stub* et un *skeleton*. Il s'agit là de deux classes d'objets, l'une respectivement sur le poste utilisateur, l'autre sur le serveur. Le *stub* et le

skeleton sont chargés de procurer à l'utilisateur client, respectivement au serveur une interface commune et multi-plateformes, qui délègue les services communications. Cette interface définit les méthodes déléguées que l'utilisateur client souhaite appeler sur le serveur. Le flux de données (voir schéma ci-dessous d'une description objet XML illustrative, sous forme de diagramme de classes) est entièrement encapsulé par ces deux classes.



20 Dans la FIG. 4 des dessins annexés est représenté, sous forme d'un diagramme de séquence illustratif et non limitatif, le fonctionnement du *stub* et du *skeleton* dans une application client/serveur lorsque la méthode 1 de l'interface objet X (voir ci-dessus) est invoquée par l'utilisateur client. Les 25 pointillés représentent la communication sur le réseau.

Il convient de noter que la fermeture de la socket est à l'initiative du client, lorsqu'une ou plusieurs méthodes ont été appelées.

30 A titre d'exemple, un utilisateur peut s'abonner ainsi à un service de location de logiciels. L'agent utilisé dialogue alors avec l' *Intowan Resource Center* , qui lui envoie les composants associés à ce service: distribution du logiciel, retrait du logiciel et contrôle. L'utilisateur peut immédiatement utiliser le logiciel chargé. La traçabilité mise en place par le système selon l'invention fournira par ailleurs des informations liées à son utilisation, telles que par exemple sa durée d'utilisation.

35 Techniquement, on met ainsi avantageusement en oeuvre:

- d'une part, un protocole approprié, dénommé ici en abrégé IVAP (Intelligent Versatile Access Protocol), qui assure un lien interactif assimilable à un cordon ombilical reliant un IRC et/ou des IRC fils à des agents, et

5 - d'autre part, un système assurant la traçabilité des biens numériques qui sont véhiculés au moyen de ce protocole.

La traçabilité, au sens où on l'entend selon l'invention, est illustrée dans la FIG. 5, dans laquelle un bien numérique 20 est packagé ou emballé en un paquet 22 grâce à une application 21 et pourvu d'une "étiquette électronique", puis déposé sur un IRC 1 et routé selon un protocole 5 vers l'IRC 23 responsable des postes finaux cibles 24,25. Une notification de réception 26 est envoyée au serveur émetteur. Les agents 7 des postes finaux, qui récupèrent le bien numérique, envoient des notifications 27,28 à leur IRC.

En pratique, le système pour la mise en oeuvre de la traçabilité susdite 15 peut comporter, à titre d'exemple, (voir FIG. 5):

- des moyens pour packager ou emballer un bien numérique 20 (tel qu'un logiciel), par exemple grâce à l'application Packer for Windows 21;
- des moyens pour munir le paquet 22 d'une "étiquette électronique logicielle" pouvant comprendre notamment les éléments suivants:

20 - nom, taille et date du paquet,
- description du contenu,
- certificat électronique, etc.,

25 • des moyens pour déposer le paquet 22 sur un IRC 1 et pour le router automatiquement (ici grâce au protocole IVAP 5) vers l'IRC 23 responsable des postes finaux cibles 24,25;

- des moyens pour l'envoi par le serveur 23, dès réception du paquet, d'une notification de réception 26 (qui comprend l'étiquette électronique du paquet, le certificat du serveur, l'heure de réception du paquet, etc.) au serveur émetteur 1;

30 • des moyens pour la récupération du paquet 22 par les agents 7 des postes finaux et pour l'envoi par les agents 7 de notifications 27,28 à leur IRC; et

35 • des moyens pour que les postes finaux informent régulièrement leur IRC des opérations qu'ils effectuent sur le bien numérique concerné (par exemple effacement, lancement du logiciel, etc.); et, en option,

- des moyens pour la consignation des notifications 26,27,28 dans des bases de données et/ou pour leur remontée automatique à tous les niveaux, de préférence jusqu'au niveau du serveur principal 1.

Il faut noter que, dans ce système, un agent peut être connecté à 5 plusieurs serveurs IRC.

Dans une forme de réalisation préférée, le système pour la distribution dynamique de données et/ou de services selon l'invention comporte des moyens de traçabilité essentiellement constitués par:

- une application 21 assurant le packaging d'un bien numérique 20 pour 10 constituer un paquet 22 et l'allocation à celui-ci d'une étiquette électronique logicielle;
- des moyens pour le dépôt dudit bien numérique sur un IRC 1;
- des moyens pour le routage du dit bien numérique selon un protocole 5 vers l'IRC 23 responsable de postes finaux cibles 24,25;
- des moyens de fourniture d'au moins une notification relative au paquet 22 et d'envoi de celle-ci au serveur émetteur 1 et/ou 6.

On entend ici par «notification relative au paquet» 22 aussi bien une notification de réception 26 et/ou 27,28 que des notifications respectivement de copie, d'effacement, de modification, ou autres.

20 L'invention a ainsi également pour objet un système pour la distribution dynamique de données et/ou de services tel que décrit plus haut, et dans lequel la distribution est effectuée avec traçabilité, par des moyens et/ou des éléments de système tels que décrits ci-dessus.

25 Dans ces formes de réalisation, le système intègre une mécanique évoluée permettant d'offrir une traçabilité complète des biens numériques traités par le système. Cette mécanique évoluée peut notamment comporter des moyens destinés à procurer des marquages appropriés des fichiers, des codes de retours, des routages, etc.

30 Sous l'angle du procédé utilisé pour sa mise en oeuvre, la traçabilité, au sens où on l'entend selon la présente invention, fait intervenir les étapes suivantes:

- packaging ou emballage d'un bien numérique 20 (tel qu'un logiciel), par exemple grâce à l'application Packer for Windows 21;
- fourniture au paquet 22 d'une "étiquette électronique logicielle" 35 pouvant comprendre notamment les éléments suivants:

- nom, taille et date du paquet,
- description du contenu,
- certificat électronique, etc.,

- dépôt du paquet 22 sur un IRC 1 et routage du paquet 22

5 automatiquement (ici grâce au protocole IVAP 5) vers l'IRC 23 responsable des postes finaux cibles 24,25;

- envoi par le serveur 23, dès réception du paquet, d'une notification de réception 26 (qui comprend l'étiquette électronique du paquet, le certificat du serveur, l'heure de réception du paquet, etc.) au serveur émetteur 1;

10 • utilisation du paquet 22 reçu par les agents 7 des postes finaux et envoi par les agents 7 de notifications 27,28 à leur IRC; et

- information régulière de leur IRC par les postes finaux des opérations qu'ils effectuent sur le bien numérique concerné (par exemple effacement, lancement du logiciel, etc.); et, en option,

15 • consignation des notifications 26,27,28 dans des bases de données et/ou remontée automatique des dites notifications à tous les niveaux, de préférence jusqu'au niveau du serveur principal 1 ou d'au moins un serveur fils 6.

20 L'invention a ainsi également pour objet un procédé pour la distribution dynamique de données et/ou de services tel que décrit plus haut, et dans lequel la distribution est effectuée avec traçabilité, par des moyens tels que décrits ci-dessus.

25 Ainsi, selon la présente invention, on réalise une plate-forme de distribution interactive de biens numériques et de services, de préférence en association avec des moyens de traçabilité.

30 Le système et le procédé selon l'invention mettent avantageusement en oeuvre une sélection préférée de moyens techniques qui se sont avérés permettre d'optimiser les résultats. Parmi tous les moyens sur lesquels a porté la sélection, on peut citer, à titre d'exemples purement illustratifs et non limitatifs:

35 - utilisation des langages objet. Ceux-ci offrent en particulier l'avantage de reposer sur des codes plus structurés, d'offrir une possibilité de réutilisation des composants mis en oeuvre, et une maintenance plus aisée et plus fiable du logiciel par rapport à d'autres langages, tels que des langages de type structuré, par exemple.

choix d'un langage de développement indépendant pour la plate-forme et les agents respectivement. En effet, il est apparu que les différences existant entre les rôles, les fonctionnalités et les caractéristiques de l'IRC et des agents rendaient préférable d'appréhender la question d'un langage de développement de façon indépendante pour chacun, tout en s'attachant à conserver une certaine cohérence, du fait que ces deux types d'unités sont destinés à communiquer entre eux.

5 Pour le développement de l'IRC, le choix s'est porté sur le langage Java. Celui-ci présente deux avantages techniques essentiels:

10 • il est multi-plateforme, c'est-à-dire qu'il s'adapte sur la majorité des systèmes d'exploitation actuellement disponibles;

- ses composants sont inter-opérables et réutilisables.

Pour le développement des agents, la préférence a été donnée au langage C++, mais les langages Java et C peuvent également être utilisés, 15 entre autres. Cependant il s'est avéré que les fonctionnalités différentes liées aux parties client de la solution de communication selon l'invention rendaient préférable un choix de langage différent pour leur développement. L'utilisation du langage Java induirait l'existence d'une Java Virtual Machine sur chaque poste client. Par ailleurs, il est hautement avantageux de 20 concevoir des programmes occupant un espace disque suffisamment réduit pour qu'ils puissent être installés sur l'ensemble des postes finals. Enfin, l'utilisation du langage C++ s'est avérée permettre une exécution plus rapide des programmes sur les postes finaux.

Pour rendre le fonctionnement du système plus dynamique, on peut 25 également opter pour des objets COM, dans le cas de l'Agent for Windows, ce choix pouvant être différent pour d'autres agents conçus pour d'autres types d'architectures. L'utilisation des objets COM permet en effet d'ajouter dynamiquement des modules aux agents et leur procure la capacité d'interagir avec leur environnement.

30 En relation avec les choix préférés susdits préconisés pour le développement des principaux acteurs du système selon l'invention, on recommande d'opter, pour la communication entre l'IRC et les agents, préférentiellement pour le protocole IVAP, lequel est fondé sur le langage XML. Le langage standard XML permet d'échanger et de stocker des 35 données au niveau de la configuration des agents, ainsi que d'améliorer la

lisibilité des données et de faciliter les évolutions souhaitées. Le protocole IVAP fondé sur ce langage, qui lui confère les propriétés avantageuses susdites, est un protocole à la fois fiable et performant.

Un perfectionnement supplémentaire, décrit plus loin, de ce système 5 selon l'invention a été réalisé pour mieux répondre à l'accroissement considérable du nombre de solutions dédiées de nos jours au transport de données, afin de permettre aux informations véhiculées de pouvoir bénéficier aisément de services à haute valeur ajoutée.

Le langage XML est idéal pour l'échange de messages et est ainsi 10 devenu un standard de l'industrie dans l'EDI (Electronic Data Interchange). Il est cependant relativement mal adapté au transport de données: il n'offre pas de possibilité de compression, ni de reprise sur erreur.

L'utilisation de composants standard appropriés et du protocole IVAP 15 permet d'adoindre des services à très haute valeur ajoutée aux informations véhiculées, en intégrant cryptage (couche de cryptage SSL (Secure Socket Layer)), reprise sur erreur, ordonnancement des opérations, traçabilité des données, entre autres.

Un tel protocole de communication entre le serveur et les agents, éventuellement par le relais d'IRC fils, est un protocole à part entière, 20 fonctionnant au-dessus des protocoles TCP/IP. Ce protocole avantageusement mis en oeuvre selon l'invention possède ainsi ses propres ports IP.

Ce protocole est conçu pour gérer des flux binaires, transactions et 25 appels de procédures distantes. Il permet de distribuer dynamiquement, sans interruption des services existants et sans intervention physique sur les plates-formes et les postes clients, des services additionnels sur la totalité ou, en option, une partie du réseau de distribution.

Le transport de données numériques et d'informations associées au moyen du système selon l'invention s'effectue sur la "Chaîne Logistique 30 Numérique" (CLN), encore dénommée "Supply Chain Numérique".

Dans un mode de réalisation, le protocole ainsi mis en oeuvre intègre la gestion d'un canal d'échange de données binaires, ce qui lui procure une meilleure fiabilité, notamment en cas d'interruption du transfert de données. Cela offre la possibilité d'effectuer par exemple des reprises sur erreur. Ainsi,

lorsqu'il est interrompu, le transfert reprend à l'endroit précis où il s'était interrompu.

5 Cette gestion des flux binaires s'effectue en natif. Elle permet notamment au protocole d'optimiser les échanges, grâce à l'adjonction optionnelle de techniques de compression, de type connu.

En langage XML, le protocole de communication utilisé présente un caractère évolutif, ce qui constitue un avantage. Cela lui fournit la capacité d'organiser des échanges de messages de façon standard et ouverte.

10 Pour sa sécurisation, on peut lui adjoindre un protocole de sécurité standard, dénommé "Secure Socket Layer" (SSL). Dans cette forme de réalisation, le protocole de transfert/gestion des données intègre une couche de cryptage SSL, qui permet de rendre les messages illisibles pendant leur acheminement.

15 L'invention est maintenant décrite plus en détail ci-après, en référence à des formes de réalisation illustratives, qui ne limitent en aucune manière l'invention revendiquée.

Le serveur et les agents de la "Chaîne Logistique Numérique" Intoan

20 Le système repose sur une architecture client/serveur. Le serveur est dénommé *Intoan Resource Center*.

L'*Intoan Resource Center* (ou *IRC*) est la plate-forme de communication centrale de ce système de communication. Il dirige des agents, qui vont diffuser les données vers les postes finals ou des *IRC* fils, qui serviront de relais vers des agents.

25 La communication entre l'*IRC* et ses agents repose préférentiellement sur un protocole dénommé *IVAP* (*Intoan Versatile Access Protocol*), protocole sécurisé créé par *Intoan*.

I Environnement

30 Le noyau *IRC-agents* constitue le cœur du système. Ces deux entités fonctionnent au sein d'un environnement faisant intervenir différentes notions et différents éléments. Il convient par conséquent de définir dès à présent ces éléments et de préciser leur place (voir Fig. 1).

1 Service

Un service est une suite structurée de transactions configurées pour répondre à un besoin technique particulier. Il est hébergé par l'IRC et 5 consommé par des agents. Le service peut également être relayé par des IRC-fils.

Les composants constituant un service peuvent être personnalisés par l'utilisateur. Ils sont organisés au sein d'une pile (stack), qui séquence leur 10 exécution. Les services actuellement disponibles permettent, entre autres, de surveiller des postes et des applications, de synchroniser des données pour les utilisateurs nomades, de louer des applications, d'effectuer des opérations de back-up, etc.

2 Chaîne Logistique Numérique

15 La *Chaîne Logistique Numérique* (CLN) assure le transport des données numériques et des informations associées entre le serveur et les postes clients. Elle permet de suivre l'acheminement des biens numériques, établissant ainsi une traçabilité sur les transports de données.

20 A titre d'exemple, lors d'un abonnement au service de location de logiciels, l'agent dialogue avec l'IRC, qui lui envoie les composants associés à ce service: distribution, retrait du logiciel et contrôle.

La traçabilité de bout en bout ainsi mise en place par le système 25 fournit des informations liées à l'utilisation de ce service, comme son temps d'utilisation par exemple. Elle permet d'apporter de la transparence sur le déroulement du processus ; il est par exemple possible de localiser une application à un moment donné, de déduire des informations sur la congestion du réseau, sur l'utilisation des ressources, sur l'acheminement 30 des données, etc. L'ensemble de ces informations remontent vers l'*Intégrated Resource Center* lors de la circulation des données numériques, grâce à l'intégration d'une mécanique évoluée : marquage de fichiers, codes de retours, routage, etc.

II Description technique générale / Architecture**1 Description****5 a Intoan Resource Center**

L'Intoan Resource Center est le serveur qui héberge les services et pilote les agents, éventuellement par l'intermédiaire des IRC-fils qui lui serviront de relais, si on le souhaite.

10 b Agent for Windows

L'Agent for Windows intervient dans la partie client du système. Cette est adaptée au système Microsoft Windows®. Il réside sur les postes clients pour diffuser des données. Il dialogue avec l'IRC qui le pilote, son *Resource Center* « père ».

15 D'autres types d'agents fonctionnant sur d'autres plate-formes (Mac®, Unix®, PalmOS®, etc.) peuvent y être ajoutés.

c Protocole de communication : l'IVAP

20 La communication entre *IRC* et agents est assurée par un protocole de communication fiable, et de préférence sécurisé : *l'Intoan Versatile Access Protocol*.

Ce protocole intègre le langage XML, ce qui le rend structuré et évolutif. Il gère flux binaires, transactions et appels de procédures distantes.

25 L'aspect sécurisé de l'IVAP et par extension, de l'ensemble des transferts effectués via ce protocole, repose sur l'utilisation d'un protocole de sécurité standard: le Secure Socket Layer (SSL).

30 Ce protocole garantit des communications privées et authentifiées sur les réseaux. Il devient par conséquent possible de transmettre des informations confidentielles sans risque qu'elles soient interceptées par un tiers. Les deux principaux navigateurs Web existants supportent ce protocole.

Le principe de fonctionnement de SSL est l'utilisation d'un couple clé privée/clé publique pour crypter les données transférées pendant la connexion SSL.

d Admintool

Outre ses parties serveur et clients, le système selon l'invention comporte un programme destiné à la configuration et à l'administration de l'IRC : l'*Admintool*. Celui-ci est livré par défaut avec la solution.

5

L'*Admintool* (ou administration tool, outil d'administration) est un programme développé en Java, permettant de configurer l'IRC à partir de n'importe quel poste connecté au réseau directement ou indirectement, par l'Internet par exemple, et d'agir sur les composants installés. Ce programme 10 se connecte sur le port GUI du serveur. L'*Admintool* configure notamment les paramètres suivants:

- le nom du serveur ;
- l'adresse IP du serveur ;
- le port destiné aux agents ;
- le port d'administration ;
- le mot de passe administrateur ;
- la cryptographie utilisée (SSL, MD5, etc.) ;
- l'emplacement des fichiers certificats (Keystore et Truststore) ;
- la connexion de la base de données à laquelle se connecter.

20 L'*Admintool* rend ainsi plusieurs composants disponibles, pour optimiser le fonctionnement du système :

- un composant de configuration des paramètres globaux d'un *Resource Center*,
- un composant d'interrogation de l'IRC sur son état courant (ce composant est accessible par deux implémentations distinctes : version XML et version Objet Java sérialisée),
- un composant d'installation et de visualisation des modules présents sur le *Resource Center*,
- un composant d'installation et de visualisation des licences agent sur 30 l'IRC.

2 Architecture

IRC et Agent for Windows sont construits autour de la même architecture. Les couches de base intégrées de part et d'autre permettent :

- à l'IRC de communiquer avec ses agents, par échanges de messages et échanges binaires (couche *IVAP*),
- le cryptage de ces transferts de données grâce à une couche SSL,
- la compression (couche *zlib*),
- 5 • l'ordonnancement des tâches,
- l'accès à des bases de données externes,
- les ajouts et suppressions de modules,
- etc.

10 Pour des questions de performance et d'organisation, les *Intowan Resource Centers* ont la faculté de pouvoir se chaîner les uns aux autres. Il devient dès lors très simple de déployer une arborescence d'IRC, chaque serveur s'identifiant et se configurant alors de façon automatique. Cela confère

15 Par ailleurs, l'IRC dispose de fonctions de réplication (mirroring) et de clustering permettant d'augmenter la disponibilité de services et de répondre à des problématiques de gestion à grande échelle des clients dans des environnements distribués.

a **Framework**

20 Le système repose sur une architecture distribuée caractérisée par sa modularité. Elle se compose en effet d'un framework et de modules.

Au sein de cette architecture modulaire, le framework intègre les couches de base et permet l'ajout dynamique de modules.

25 b **Modules**

30 Un module est un groupe de composants métier, intégrant un ensemble de services, qui permet d'effectuer par exemple le déploiement d'applications, la surveillance de machines, la maintenance à distance, la téléconfiguration d'un poste, la télémaintenance de progiciels ou la sauvegarde de données utilisateurs, etc.

Installation de modules

Un module se matérialise par un fichier JAR (Java Archive), disponible sur cédérom ou directement sur Internet.

5 Ce fichier JAR contient:

- un fichier XML de description du module,
- des classes pour les composants,
- des bibliothèques DLL pour l'Agent For Windows,
- 10 • un script SQL pour créer ou modifier des tables,
- des scripts SQL pour créer des services génériques (services proposés par défaut) dans la base.

15 Un module s'installe via l'*Admintool* qui effectue un upload sur le *Resource Center*. Celui-ci se charge alors de le stocker correctement sur son disque et de valider le numéro de licence transmis au même moment.

20 Il existe deux catégories de modules : les *core modules*, qui sont les modules installés par défaut, et les *business modules*, qui seront installés en fonction des besoins particuliers des utilisateurs. Ces derniers peuvent personnaliser eux-mêmes leur système, en ajoutant les modules dont ils ont besoin.

Core modules

25 Les *core modules* sont nécessaires au fonctionnement du système. Ils sont au nombre de 6 : Framework, Subscribe, Identify, File Transfer, Process et Data Recipients.

Module Framework

Le module *Framework* contient les composants suivants :

30 **Composant Scheduler**

Ce composant permet de programmer dans le temps des séquences d'appels d'autres composants. Il est intensivement utilisé par l'agent pour le lancement à intervalles réguliers d'appels de transactions et de leurs composants associés.

Composant Parser

Le composant *Parser* intègre des fonctions de codage/décodage du langage XML. Il accepte en entrée une structure de données de type DOM pour générer un flux XML, mais ne crée que des flux XML qui sont intégrés 5 au sein du protocole IVAP.

Dans la mesure où son utilisation est omniprésente dans les échanges *IRC-agent*, ce composant est intégré au module *Framework*. Il est présent à la fois dans l'*IRC* et dans l'*agent*.

10 Composant Logger

Logger lit et écrit le flux XML qui lui parvient dans une base de données. Ce composant se trouve côté *agent*.

Composant Stacker

15 Le composant *Stacker* délivre les séquences *Business Logic* en attente sur l'*Intégrated Resource Center* pour chaque *agent*.

Une *stack* est une structure de données, constituée d'une séquence d'exécution et de ses paramètres, appartenant à un service et destinée à programmer un *agent* en fonction du service sélectionné.

20 Ce composant permet à l'*IRC* de contacter directement un *agent* pour lui demander de faire immédiatement appel à sa *stack*, ce qui est utile pour exécuter une transaction urgente (déploiement immédiat d'une application sur un poste local par exemple). Par ailleurs, ce composant est également responsable de la préparation de la *stack* des *agents* en fonction des directives envoyées par l'*Adminintool*.

25 En ce qui concerne l'*agent*, le composant *Stacker* permet l'ordonnancement de la *Business Logic* d'une transaction dans le temps.

Module Subscribe

30 Ce module ne comporte qu'un composant, appelé *Assign Service*. *Assign Service*, installé à la fois côté serveur et client, permet à un *agent* de demander la liste des services proposés par un fournisseur, mais également d'inscrire un *agent* à un service d'un *IRC*.

35 L'*IRC* envoie la liste exhaustive des services qu'il désire fournir aux *agents* qui lui sont rattachés.

Module Identify

Comme le module *Subscribe*, ce module ne comporte qu'un seul composant, *Identify*, qui permet d'identifier la consommation d'une transaction par un agent sur un *Resource Center*. Présent côté *IRC* et côté agent, il est généralement utilisé en début de chaque transaction.

L'*Intégration Resource Center* valide l'identité de l'agent qui effectue la transaction et vérifie qu'il est autorisé à utiliser cette transaction. Dans l'éventualité où l'agent qui se présente est inconnu ou n'est pas inscrit à un service, l'*IRC* n'autorise pas la transaction. Cependant, si un tel agent se présente pour s'inscrire à un service, l'*IRC* le référence dans sa base de données.

Lors de son installation, un nouvel agent se crée un identifiant unique qui lui permet de s'identifier au sein de chaque *IRC* sur lequel l'agent veut consommer un service.

15

Module File Transfer

Le module *File Transfer* prend en charge les méthodes de téléchargement de fichiers avec reprise sur erreur. Chacun des composants constituant ce module est situé à la fois côté *IRC* et côté agent.

20

Quatre composants, complémentaires, constituent ce module:

Composant Download

Ce composant accueille les demandes de download des agents. La particularité du système réside dans son fonctionnement en mode reprise sur erreur; ainsi l'agent peut demander le download d'un fichier à un endroit précis.

Composant Upload

Ce composant prend en charge les méthodes d'upload de fichiers avec reprise sur erreur. Comme évoqué dans le cas du composant *Download*, la particularité du système est de fonctionner en mode reprise sur erreur, ce qui permet à l'*IRC* de demander à l'agent l'upload d'un fichier à un endroit précis. Le composant stocke les fichiers sur le *Resource Center* selon sa configuration locale et les paramètres transmis par l'agent.

Composant Pack

5 Le composant *Pack* effectue le groupement et la compression de fichiers en format *Intowan.pak*. Il accepte une expression comme paramètre ; cette expression permet de définir les fichiers à compresser. Il est possible de spécifier des options d'archivage et de compression des fichiers.

Composant Unpack

10 Le composant *Unpack* effectue la décompression des fichiers *pak*, et cela, en se basant sur les règles de déploiement incluses dans le fichier compressé. Ces règles concernent les suppression, copie de fichiers, chemins de fichiers, etc.

15 Module Process

Le module *Process* est constitué de trois composants, tous situés côté *IRC*.

Composant Filter

20 *Filter* permet d'extraire des propriétés sur un composant pour les transmettre à un autre composant.

Composant Format

25 Le composant *Format* permet de présenter, ou formater, à un autre composant une chaîne de caractères dans laquelle on va trouver des variables passées en entrées.

Composant External Connect

30 Le rôle de ce composant consiste à connecter une transaction sur une application externe. Il agit comme un système ayant une architecture client/serveur, sur lequel se connecte le processus externe. Il est par exemple possible d'utiliser *External Connect* pour connecter une transaction de déploiement d'application à une solution de facturation externe.

Module Data Recipients

De même que pour le module *Process*, l'ensemble des composants constituant le module *Data Recipients* se trouvent côté *Internet Resource Center* et ne sont utilisés que dans cette partie du système.

5 Le module *Data Recipients* comporte 6 composants :

Composant Write to Database

Ce composant prend en charge la connexion sur tous les types de bases de données.

10

Composant Write to File

Write to File permet d'écrire dans un « fichier plat » un flux de données provenant de composants. Ce composant est utile pour constituer des fichiers de type LOG.

15

Composant Send Mail

A la manière du composant *Write to Database*, *Send Mail* permet d'écrire un flux de données provenant de composants dans un e-mail et de l'envoyer à son destinataire.

20

Composant Play Sound

Dans le cadre d'un service, le composant *Play Sound* permet d'émettre un son sur l'*IRC* pour signaler un événement. Il est bien entendu nécessaire que le serveur hébergeant le *Resource Center* soit configuré pour émettre des sons.

25

Composant Send Snmp Trap

Placé au sein d'un service, ce composant permet d'envoyer un trap SNMP vers une solution de gestion de parc afin de remonter une alerte sur la console de supervision d'un logiciel tiers.

30

Composant Write to Ldap

Le composant *Write to Ldap* prend en charge la connexion sur une base LDAP (base utilisant le protocole de gestion d'annuaires LDAP).

35

Business modules

Son aspect modulaire fournit une caractéristique propre au système selon l'invention son extensibilité, dans la mesure où il est possible d'ajouter des modules à la configuration de départ, en fonction de l'usage de postes clients particuliers ; ces modules sont dénommés *business modules*.

5 Les agents et l'*Intalan Resource Center* supportent un nombre illimité de modules. De plus, ceux-ci sont ajoutés sans que la moindre interruption de service ne soit nécessaire.

10 En termes de programmation, l'ajout de modules fait intervenir des objets COM ; ceux-ci permettent en effet de créer des instances objets avec des bibliothèques.

15 Il existe actuellement 4 business modules, auxquels peuvent être adjoints d'autres modules spécifiques *Application Distribution*, *Application ASP*, *Optimail* et *Assets Sourcer*. Les composants constituant chacun de ces modules sont tous installés côté *IRC* et côté agent, assurant ainsi la communication entre les deux entités.

Module Application Distribution

20 Le module *Application Distribution* regroupe les composants et services qui traitent du déploiement d'applications et de la distribution de données. Il comporte deux composants :

Composant Deploy Applications

25 *Deploy Application* permet le déploiement d'applications (préparées avec le *Winpacker*) sur les postes équipés d'un *Agent for Windows*. Sur l'*IRC*, il avertit celui-ci de l'état d'une installation.

Composant Remove Applications

30 Fonctionnant sur le même principe que le composant de déploiement, *Remove Application* effectue le retrait des applications sur les postes équipés d'un *Agent for Windows*.

Module Application ASP

Ce module est constitué de composants dédiés respectivement à la surveillance et à la facturation des applications utilisées.

5 **Composant Monitor Applications**

Le composant *Monitor Applications* permet d'effectuer la surveillance d'applications installées par le composant de déploiement d'applications. Il est également chargé de collecter les informations relatives à l'usage journalier d'une application.

10

Composant Bill Applications

Ce composant permet la facturation des applications installées par le composant *Application Deployer* ainsi que celle de l'usage journalier d'une application par un utilisateur.

15

Module Optimail

Le module *Optimail* permet d'optimiser la gestion de la messagerie. Les composants constituant ce module sont complémentaires.

20

Composant Read Mailbox

Ce composant permet la connexion d'un agent sur un serveur de messagerie de type POP ou IMAP afin de relever la liste des messages en attente. La partie *IRC* de ce composant peut relever la liste des messages en attente dans le but, par exemple, de nettoyer une messagerie des messages trop volumineux qu'elle contient.

25

Composant Read Mail

Installé côté *IRC*, le composant *Read Mail* permet de lire les messages sélectionnés dans la liste fournie par le composant *Read Mailbox*.

30

Composant Store Mail

Le composant *Store Mail* copie le contenu du message lu sur l'*Infoan Resource Center*, de façon à ce que ce message soit disponible hors connexion Internet.

35

Composant Send Mail

Send Mail permet l'envoi d'un message, étant ainsi complémentaire au composant *Read Mail*.

5 **Composant Clean Mailbox**

Ce composant permet d'effectuer un nettoyage du serveur en une seule manipulation. Il sélectionne les messages à effacer en fonction des critères spécifiés par l'utilisateur (taille du message, date de réception, etc.).

10 **Module Assets Sourcer**

Le module *Assets Sourcer* remonte des données bas niveau concernant le poste sur lequel l'agent est installé. Il est constitué de deux composants :

15 **Composant Get Assets List**

Ce composant collecte les données relatives aux postes hébergeant l'agent : type de microprocesseur, fréquence, espace disque, etc.

Composant Store Assets List

20 Le composant *Store Assets List* permet à l'*Intaoan Resource Center* de stocker les données recueillies par le composant *Get Assets List*.

3 Installation25 a **Installation du Resource Center**

L'*Intaoan Resource Center* est principalement distribué sur cédérom. Développé en Java, il est conçu pour fonctionner sur les plate-formes pouvant supporter une Java Virtual Machine (version 1.3 et supérieures) :

- Windows 95, 98, Me, NT 4.0, 2000 (NT et 2000 étant recommandés),
- Solaris SPARC 2.5 et versions supérieures,
- Mac OS 8.0 et versions supérieures,
- HP-UX 10.x, 11.x et versions supérieures,
- AIX 4.x et versions supérieures,
- BSD, Linux Intel, IRIX, Tru64 et autres plates-formes UNIX.

Il est également nécessaire que le serveur destiné à accueillir l'IRC dispose d'une base de données préalablement à l'installation.

Un programme installe les composants et les bibliothèques de l'*Intoaan Resource Center* sur le système cible.

5 En fonction du système d'exploitation sous lequel le programme d'installation est lancé, certains paramètres, tels que le répertoire d'installation sous Windows, seront prérenseignés. Un programme de désinstallation, ainsi qu'une documentation de dernière minute sont également copiés dans le répertoire d'installation.

10

b Installation de l'Agent for Windows

15 L'installation de l'*Agent for Windows* s'effectue depuis un programme d'installation de type InstallShield. Le remplacement de ce programme, développé en C++, par le système d'installation interne d'*Intoaan Technology* est envisageable à l'avenir.

L'*Agent for Windows* s'installe par défaut dans le répertoire « c:\Program Files\Intoaan\Agent ». Un programme de désinstallation et une documentation récente apparaissent également dans ce répertoire.

20 Une fois l'agent installé, le menu *Démarrer* (Programmes donne accès au dossier Intoaan Technology, qui contient les éléments suivants :

- *Intoaan Agent*, raccourci vers IAgent.exe (l'agent est accompagné d'un fichier de configuration en XML) ;
- *Uninstall* (raccourci vers unwise.exe ou autre : programme de désinstallation) ;
- *Fichier Readme* (raccourci vers les informations de dernière minute).

4 Identification

30 Lors de la première utilisation, l'agent vérifie qu'il possède un identifiant (id_agent) dans son fichier de configuration. Si tel n'est pas le cas, l'agent va générer un identifiant unique en utilisant une clé de 50 caractères. Cette clé sera valable tout au long de la vie de l'agent.

L'adresse IP du serveur permet la connexion d'identification de l'agent, qui transmettra son identifiant unique ainsi que les informations utilisateurs communes (nom, prénom, adresse, etc.).

5 Abonnement aux services

Le serveur de chaque prestataire dispose d'un certain nombre de services auxquels l'utilisateur, à travers l'agent, peut souscrire. Chaque abonnement d'un agent à un service consomme une licence. Chaque *Intoan Resource Center* est vendu avec une licence multiservices pour dix agents.

5 Parmi les services proposés, sont actuellement disponibles les services de déploiement d'applications, de synchronisation de fichiers et/ou de répertoires, de récupération de données, de sauvegarde, d'abonnement et de résiliation d'abonnement à d'autres services, etc.

10 L'abonnement peut s'effectuer soit de manière directe depuis l'interface de l'agent, soit par une directive stack qui appellera une transaction d'abonnement. Une directive stack décrit une ligne de la stack.

15 Certains services proposés à l'abonnement sont gratuits, d'autres payants. Les informations relatives à la tarification sont accessibles dans la page de présentation de chaque service.

a Licences - Attribution des licences

Au cours de l'installation d'un module, l'utilisateur doit entrer un code de licence valide.

20 Ce code est à saisir dans l'interface de l'*Admintool* dans l'espace prévu à cet effet.

25 Une licence peut être collectée à partir d'un fichier texte (envoyé par courrier électronique par exemple). Le code de licence n'excède pas 32 caractères et n'inclut pas de signe diacritique (accents, cédille, etc.).

Procédure d'attribution de licence(s)

30 L'utilisateur sollicite l'adresse IP du serveur sur lequel il souhaite installer son *Intoan Resource Center*. Une licence cryptée est créée; elle contient les informations dont l'IRC a besoin et se décompose, par exemple, comme suit:

[NUMERO DE SERIE | SKU | NB de LICENCES | ADDRESSE IP]

Au démarrage et à intervalles réguliers (par exemple toutes les heures), l'IRC vérifie la concordance entre ses paramètres de configuration (adresse IP) et les licences installées.

b Annuaire des services

Pour que les agents puissent facilement s'abonner à des services, ils doivent pouvoir se référer à un serveur principal recensant tous les services disponibles. De même, les services locaux doivent être recensés au niveau de 5 l'annuaire local pour que les agents puissent s'abonner à ces services sans pour autant nécessiter de connexion Internet.

Deux types d'annuaires peuvent donc être implémentés:

- *Intoan Main Service Directory (IMSD)* : l'annuaire de référence *Intoan* ; 10 dans l'interface, le nom du service est accompagné du nom du prestataire fournissant ce service ;
- *Local Service Directory (LSD)* : l'annuaire local propre à chaque *IRC*. Ces deux annuaires sont accessibles par sélection de *Nouveaux Services* dans le menu de l'écran principal.

15

Intoan Main Service Directory

L'ensemble des services publics est référencé au sein d'un annuaire global. Cet annuaire est hébergé et maintenu par *Intoan*. Dans un premier temps, une seule instance de ce serveur sera fonctionnelle au niveau 20 mondial. Il est probable que l'*IMSD* bénéficie à moyen terme d'une architecture matérielle de haut niveau (cluster, mirroring, etc.).

Local Service Directory

Chaque *IRC* possède son propre annuaire, le *Local Service Directory*. 25 Cet annuaire permet de répertorier facilement les services privés et de les rendre accessibles par les agents locaux, sans avoir à gérer de système de droits d'accès, ni nécessiter de connexion Internet.

c Inscription**Par l'interface de l'agent**

L'accès à l'*Annuaire des Services* permet de s'abonner à un ou plusieurs services, publics ou privés (stockés localement sur l'*IRC*).

Par une directive stack

5 L'abonnement d'un agent à un service particulier peut être effectué par une directive stack (il s'agit alors d'un abonnement forcé), cette directive pouvant avoir été placée dans la stack par le port GUI, pour appeler une transaction d'abonnement (transaction *Register to Service*).

d Référencement d'un nouveau service

10 Un *Intégrated Resource Center* peut demander le référencement d'un nouveau service au sein de l'annuaire global.

15 Lorsqu'un prestataire définit un nouveau service, il peut le référencer automatiquement au sein de cet annuaire. Pour cela, il doit définir les caractéristiques du service à référencer (en le décrivant notamment) ainsi que le type de référencement qu'il souhaite effectuer (publique ou privée/local).

15

e Suppression d'un service

Un *IRC* peut demander la suppression d'un service.

6 IHM de l'Agent for Windows

20 Une interface graphique de l'*Agent for Windows Simple* et intuitive permet d'accéder à l'ensemble de ses fonctionnalités.

7 Compression

25 La compression vise à optimiser les transferts de données en termes de temps et à réduire l'espace disque consacré à l'accueil des applications et des programmes sur les postes clients. La librairie utilisée dans l'agent est *zlib*. Cette librairie est un standard ; elle est notamment utilisée dans des produits tels que celui dénommé commercialement *Winzip*.

8 Programme additionnel : le Winpacker

30 Le *Winpacker*, ou *Packer for Windows*, est un programme indépendant, développé en C++, qui permet aux clients du système de préparer eux-mêmes les packages destinés à la diffusion sur les postes de leurs clients. Ce programme fait partie du "toolkit" de création de packages à destination des *Intégrated Resource Centers*. Son utilisation est à la fois simple et intuitive.

5 *Winpacker* permet de créer un fichier, qui portera l'extension *.pak*, regroupant tous les fichiers d'un répertoire ou d'une liste. Ce nouveau fichier *.pak* contient les fichiers de l'application à diffuser vers les agents, et est lié à un fichier de description XML. Enfin, ce fichier de regroupement peut être compressé.

III Choix techniques

10 Parmi les diverses options disponibles pour développer la plate-forme et les agents selon l'invention, on a choisi d'utiliser préférentiellement des langages objet. Ceux-ci offrent notamment l'avantage de codes plus structurés, d'une possibilité de réutilisation des composants et d'une meilleure maintenance du logiciel par rapport à d'autres langages, tels que des langages de type structuré par exemple.

15 Etant donné les différences existant entre les rôles, les fonctionnalités et les caractéristiques de l'*IRC* et de l'*agent*, il semblait nécessaire d'appréhender la question d'un langage de développement de façon indépendante pour chacun, tout en conservant une certaine cohérence (communication entre ces deux unités).

20 **1 Développement de l'Infoan Resource Center**

Le choix de développement de l'*IRC* s'est porté préférentiellement sur le langage Java. Celui-ci présente en effet deux principaux avantages techniques:

25

- il est multi-plateforme, c'est-à-dire qu'il s'adapte sur la majorité des systèmes d'exploitation actuellement disponibles ;
- ses composants sont inter-opérables et réutilisables.

Ces propriétés sont nécessaires dans l'optique de conception d'un serveur multi-plateforme tel que l'*IRC*, rendant ainsi ce choix technique cohérent.

30

2 Développement de l'Agent for Windows

Si Java constitue la meilleure option dans le cadre du développement côté serveur, les fonctionnalités liées à la partie client du système selon l'invention, notamment l'*Agent for Windows*, impliquent un choix différent pour son développement.

D'une part, l'utilisation du langage Java induit l'existence d'une Java Virtual Machine sur chaque poste client. Par ailleurs, il est nécessaire de concevoir des programmes occupant un espace disque suffisamment réduit pour pouvoir être installés sur l'ensemble des postes finals. Enfin, 5 l'utilisation de ce langage permet une exécution plus rapide des programmes.

Dans cette optique, la demanderesse a choisi d'utiliser préférentiellement le langage C++ dans le développement de ses agents.

Toutefois, d'autres agents pourront éventuellement être développés 10 dans d'autres langages, et ce, en fonction de l'environnement pour lequel ils sont conçus.

Un autre choix technique visant à rendre le système plus dynamique consiste à utiliser des objets COM. L'utilisation de ces objets permet en effet d'ajouter dynamiquement des modules à l'Agent for Windows et leur offre 15 la capacité d'interagir avec leur environnement.

3 Communication IRC/agent

En accord avec les choix effectués pour le développement des principaux acteurs du système, la communication entre l'IRC et l'agent 20 imposait également un choix technique; comme évoqué précédemment, ce choix s'est porté sur le langage XML. Ce langage sert de base pour le protocole de communication *Intوان Versatile Access Protocol (IVAP)*, protocole créé par *Intوان*.

Le langage standard XML permet d'échanger et de stocker des 25 données au niveau de la configuration de l'Agent for Windows, ainsi que d'améliorer la lisibilité des données et de faciliter les possibilités d'évolution.

Bénéficiant ainsi de ces propriétés, IVAP est un protocole à la fois fiable et performant.

Parmi les avantages induits par la modularité du système selon 30 l'invention, il faut noter les suivants :

- l'ajout d'un ou plusieurs modules peut être effectué sans la moindre interruption de service ;
- un nombre théoriquement illimité de modules peut être supporté par les agents et les serveurs ;
- les agents peuvent travailler de manière autonome et invisible;

- l'intégration des éléments du système permet de répondre à un ensemble de problèmes avec un même ensemble d'éléments ;
- tous les types de connexions (par exemple par modem, câble, xDSL, liaison spécialisée, entre autres) et tous les débits peuvent être 5 supportés par les éléments du système.

La traçabilité rendue possible par ce procédé permet quant à elle d'apporter une transparence exceptionnelle sur le déroulement de tout processus mettant en oeuvre le système selon l'invention.

REVENDICATIONS

1. Système pour la distribution dynamique de données et/ou de services, caractérisé en ce qu'il comporte essentiellement:
 - 5 - un serveur (1), et éventuellement des serveurs fils (6);
 - au moins un agent (7), résident ou apte à devenir résident sur un terminal utilisateur, et apte à être mis en communication avec le serveur (1) et/ou au moins un logiciel serveur fils (6); et
 - au moins une couche de communication (18) apte à véhiculer des appels de méthodes distantes sur des objets situés sur un serveur (1) et/ou un serveur fils (6), et à mettre en oeuvre au moins un protocole de communication (5) apte à assurer un lien interactif entre un serveur (1,6) et des agents (7),
tandis que ledit protocole comporte:
 - 15 - des moyens le rendant apte à produire et gérer des communications entre au moins un agent et l'IRC, notamment par le biais de messages XML;
 - des moyens pour appeler une méthode distante par lecture/écriture d'objets de types respectivement *ObjectRequest* ou *ObjetRequête* et *ObjectResponse* ou *ObjetRéponse*.
2. Système selon la revendication 1, caractérisé en ce que le ou les serveurs et les agents sont construits selon la même architecture, sur la base d'un "framework", complété par des ajouts dynamiques optionnels de modules.
25
3. Système selon la revendication 2, caractérisé en ce que les couches de base intégrées dans le serveur et dans les agents et/ou dans les serveurs fils comportent des moyens appropriés pour assurer:
 - 30 - la communication avec les agents, pour échanges de messages et échanges binaires, et en option,
 - le cryptage, et/ou
 - la compression, et/ou
 - l'ordonnancement des tâches, et/ou
 - 35 - l'accès à des bases de données externes, et/ou

- des ajouts/suppressions de modules.

4. Système selon l'une quelconque des revendications 1 à 3, caractérisé en ce qu'il comporte:

- 5 • des moyens pour packager ou emballer un bien numérique (20);
- des moyens pour munir le paquet (22) d'une "étiquette électronique logicielle" pouvant comprendre notamment les éléments suivants:
 - nom, taille et date du paquet,
 - description du contenu,
 - certificat électronique, etc.,
- 10 • des moyens pour déposer le paquet (22) sur un IRC (1) et pour le router automatiquement vers l'IRC (23) responsable des postes finaux cibles (24,25);
- des moyens pour l'envoi par le serveur (23), dès réception du paquet, d'une notification de réception (26) au serveur émetteur (1);
- 15 • des moyens pour la récupération du paquet (22) par les agents (7) des postes finaux et pour l'envoi par les agents (7) de notifications (27,28) à leur IRC; et
- des moyens pour que les postes finaux informent régulièrement leur IRC des opérations qu'ils effectuent sur le bien numérique; et, en option,
- 20 • des moyens pour la consignation des notifications (26,27,28) dans des bases de données et/ou pour leur remontée automatique à tous les niveaux, de préférence jusqu'au niveau du serveur principal (1) ou d'au moins un serveur fils (6).

25

5. Système pour la distribution dynamique de données et/ou de services selon l'une quelconque des revendications 1 à 3, caractérisé en ce qu'il comporte des moyens de traçabilité essentiellement constitués par:

- 30 - une application (21) assurant le packaging d'un bien numérique (20) pour constituer un paquet (22) et l'allocation à celui-ci d'une étiquette électronique logicielle;
- des moyens pour le dépôt dudit bien numérique sur un IRC (1);

- des moyens pour le routage du dit bien numérique selon un protocole (5) vers l'IRC (23) responsable de postes finaux cibles (24,25);
- des moyens de fourniture d'au moins une notification relative au paquet (22) et d'envoi de celle-ci au serveur émetteur (1,6).

5

6. Système selon la revendication 5, caractérisé en ce qu'il comporte des moyens pour l'envoi de notifications (27,28) par les agents (7) des postes finaux, qui utilisent le paquet reçu (22), à leur IRC.

10

7. Système selon l'une des revendications 2 ou 3, caractérisé en ce que les modules sont choisis parmi des modules du type visant à permettre le déploiement d'applications, la surveillance de machines, la maintenance à distance, la téléconfiguration d'un poste de travail informatisé, la télémaintenance de progiciels et/ou la sauvegarde des données utilisateur.

15

8. Système selon l'une quelconque des revendications 1 à 7, caractérisé en ce que le logiciel serveur est en langage Java.

20

9. Système selon l'une quelconque des revendications 1 à 8, caractérisé en ce que l'implémentation de l'agent pour le système d'exploitation d'un terminal connecté est effectuée en langage Java, C ou C++.

25

10. Procédé pour la distribution dynamique de données et/ou de services, caractérisé en ce qu'il comporte essentiellement les étapes consistant à:
 - procurer au moins un agent, c'est-à-dire un support de programme, qui peut être acquis en tant que tel ou formaté et chargé à partir d'un site Internet;
 - installer un tel agent, pour qu'il devienne résident sur la machine de l'utilisateur;
 - effectuer la souscription de l'utilisateur aux services proposés par un serveur (IRC) et/ou des serveurs fils, directement depuis l'interface de l'agent, aussi bien pour des services locaux que pour

30

des services publics, et aussi bien pour des services payants que pour des services gratuits;

- effectuer l'inscription à ces services, avec corrélativement chargement ou téléchargement automatique des composants nécessaires à leur activation, et
- en option, effectuer l'intégration par forçage de services fournis automatiquement.

5

11. Procédé selon la revendication 10, caractérisé en ce qu'il met en oeuvre un protocole dont les étapes essentielles consistent à:

- produire et gérer des communications entre au moins un agent et le serveur et/ou un serveur fils, notamment par le biais de messages XML;
- appeler au moins une méthode distante par lecture/écriture d'objets de types respectivement *ObjectRequest*, ou *ObjetRequête*, et *ObjectResponse*, ou *ObjetRéponse*, avantageusement au moyen d'un *stub* et d'un *skeleton*.

10

12. Procédé selon l'une des revendications 10 ou 11, caractérisé en ce qu'il comporte les étapes suivantes:

- packaging ou emballage d'un bien numérique (20);
- fourniture au paquet (22) d'une "étiquette électronique logicielle" pouvant comprendre notamment les éléments suivants:
 - nom, taille et date du paquet,
 - description du contenu,
 - certificat électronique, etc.,
- dépôt du paquet (22) sur un IRC (1) et routage du paquet (22) automatiquement vers l'IRC (23) responsable des postes finaux cibles (24,25);
- envoi par le serveur (23), dès réception du paquet, d'une notification de réception (26) au serveur émetteur (1);
- utilisation du paquet (22) reçu par les agents (7) des postes finaux et envoi par les agents (7) de notifications (27,28) à leur IRC; et
- information régulière de leur IRC par les postes finaux des opérations qu'ils effectuent sur le bien numérique; et, en option,

15

20

25

30

35

- consignation des notifications (26,27,28) dans des bases de données et/ou remontée automatique des dites notifications à tous les niveaux, de préférence jusqu'au niveau du serveur principal (1) ou d'au moins un serveur fils (6).

5

13. Procédé selon la revendication 12, caractérisé en ce qu'il comporte en outre les étapes selon lesquelles:

- les postes finaux informent leur IRC (1,6) des opérations qu'ils effectuent sur le bien numérique (20), et/ou
- les notifications sont consignées dans des bases de données et automatiquement remontées à tous les niveaux, de préférence jusqu'au niveau du serveur principal (1) ou d'au moins un serveur fils (6).

10

14. Utilisation du système selon l'une quelconque des revendications 1 à 9 pour la distribution d'applications et/ou de services à des utilisateurs distants connectés, sur tous types de réseaux et d'architectures distribuées, notamment pour le déploiement d'applications, la surveillance de machines, la maintenance à distance, la téléconfiguration d'un poste de travail informatisé, la télémaintenance de progiciels et/ou la sauvegarde des données utilisateur.

15

15. Utilisation du procédé selon l'une quelconque des revendications 10 à 13 pour la distribution d'applications et/ou de services à des utilisateurs distants connectés, sur tous types de réseaux et d'architectures distribuées, notamment pour le déploiement d'applications, la surveillance de machines, la maintenance à distance, la téléconfiguration d'un poste de travail informatisé, la télémaintenance de progiciels et/ou la sauvegarde des données utilisateur.

20

25

30

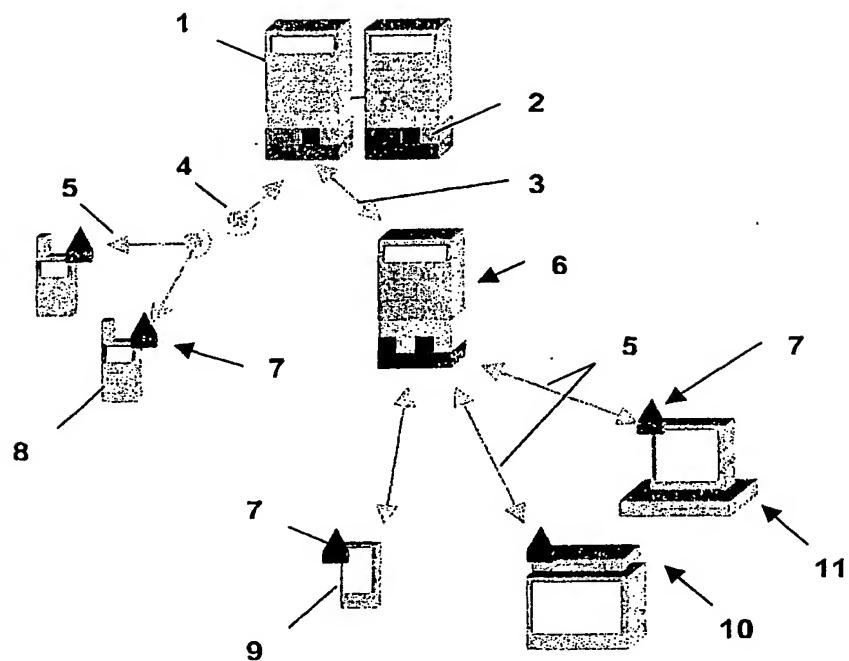


FIG. 1

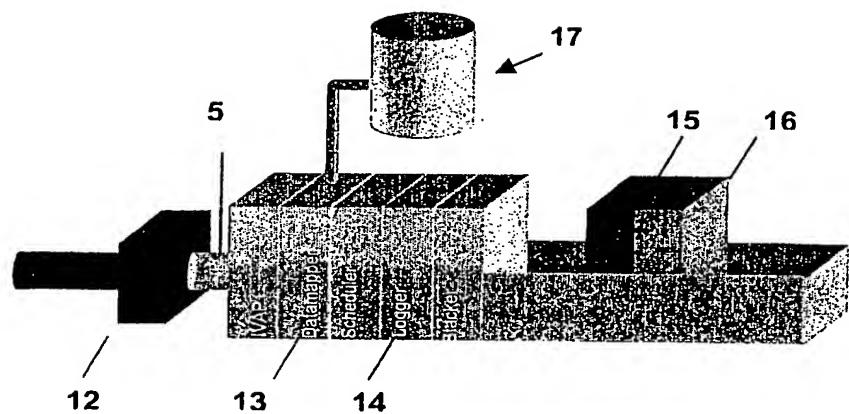


FIG. 2

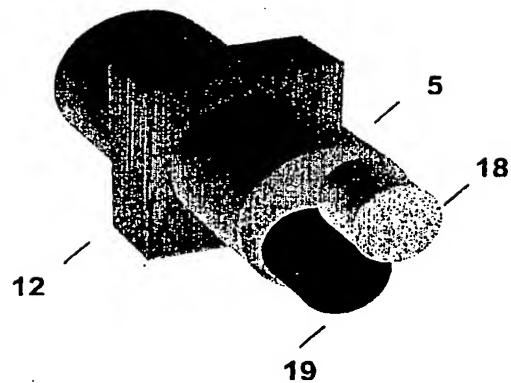


FIG. 3

3/4

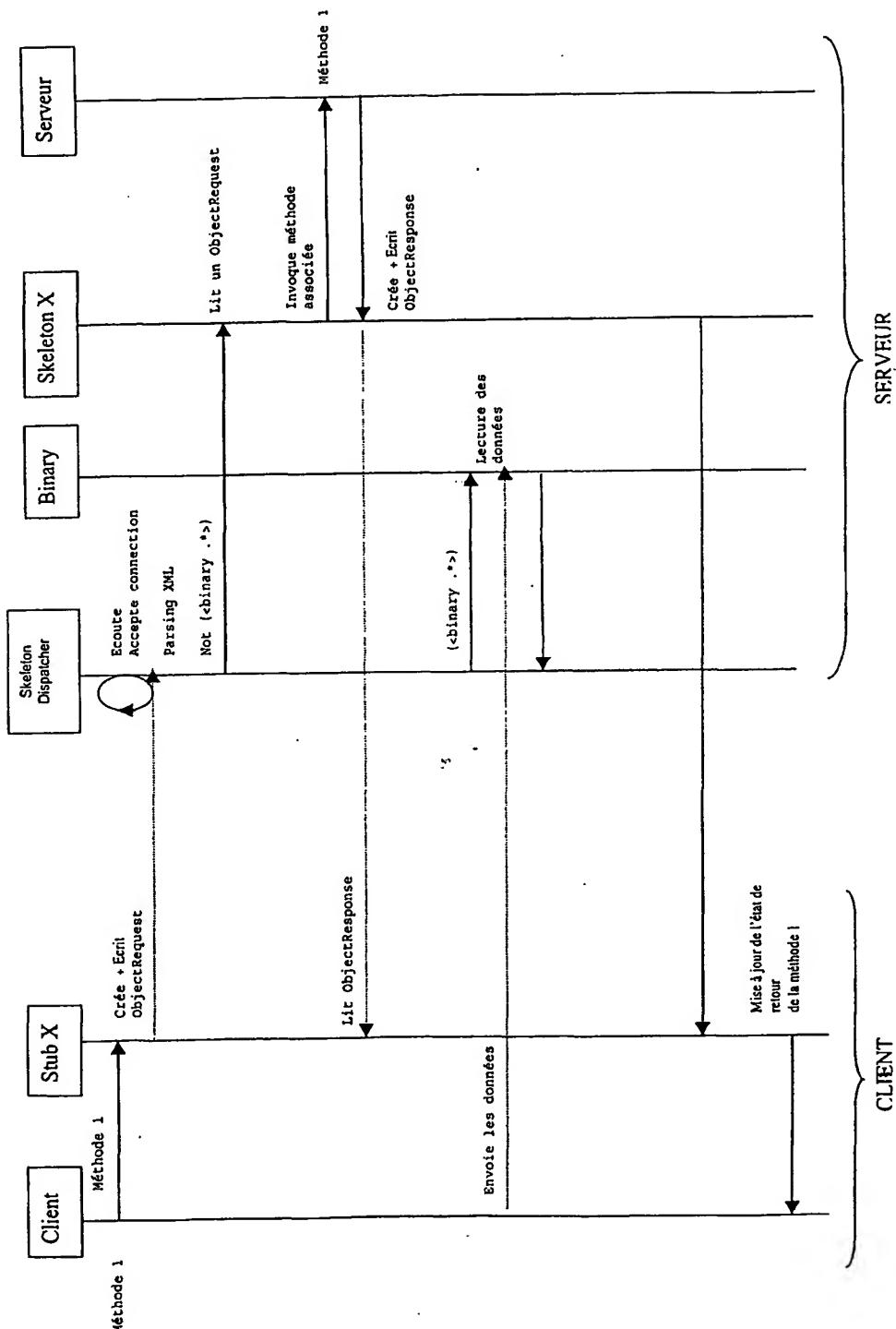


FIG. 4

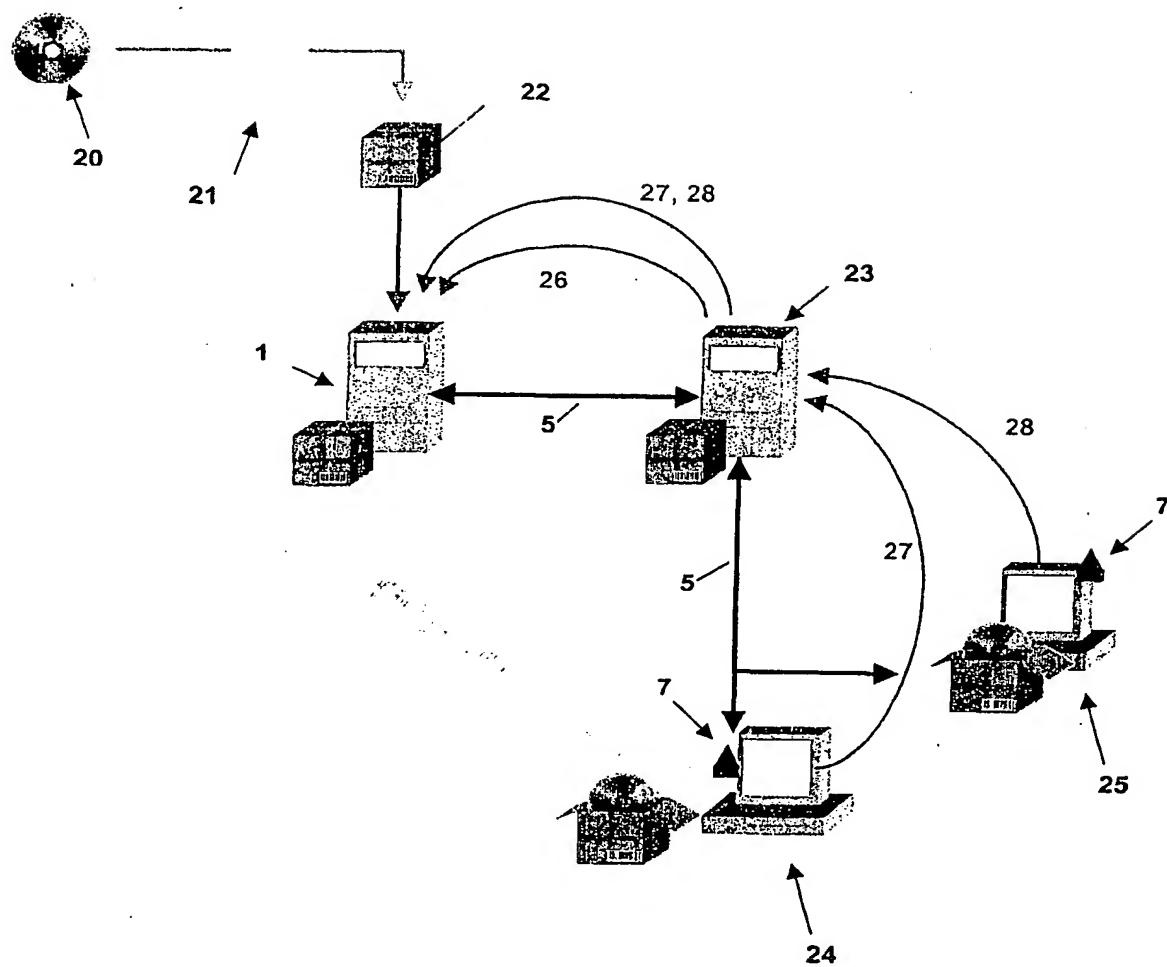


FIG. 5

THIS PAGE BLANK (USPTO)